

数据库平台

互操作性

数据库

计算机网络

(11)

52-54, 5

可互操作数据库平台的实现途径*)

唐雪飞 熊 萍 刘锦德

(电子科技大学微机所 成都610054)

TP311.13

A

摘 要 如何实现数据库互操作性,目前正在从两个方面进行努力:一是发展标准,即建立一个统一的数据库规范;二是承认数据库之间存在差异这一现实,构造可互操作数据库平台。本文将从构造可互操作数据库平台的角度,对数据库互操作性的实现进行较全面的探讨。

1 引言

所谓数据库互操作性是指在计算机网络环境中提供对异种多数据库的服务进行透明访问的一种途径^[1]。数据库互操作性不仅要求实现用户对任意异种数据库的完全透明的访问(包括查询和更新等),而且还支持不同数据库系统间的相互动作。其主要特征包括^[1]:透明多数据库访问;双向数据库访问;数据库特有功能的访问;避免资源重复和单线索操作;支持多客户平台,使用方便;提供有关远程数据库环境的信息。数据库互操作性既不是一个单纯的网络互连的问题,也不仅仅是一个分布式数据库访问的问题,它涉及的内容较广,面临的问题也较多。如何才能实现一个数据库互操作性的环境,一直为人们所关注,并正在做不懈的努力。目前主要从两个方面进行着努力:一是发展标准,即建立一个统一的数据库规范,消除由于标准的不完善而造成的各数据库系统间的差异;二是面对数据库之间存在差异这一现实,构造可互操作的数据库平台。有关发展标准的问题,在文[9]中已作了详细的讨论,在此不再赘述。本文将重点探讨可互操作数据库平台的实现途径。

可互操作数据库平台,是指在承认数据库异构这一事实的情况下,为了满足异种数据库互操作性的要求,由不同厂商或者研究组织推出的软件产品。异种数据库互操作性要求相应的软件产品能够^[1]:

- 管理访问后端数据源所需的网络设施;
- 通过对不同 SQL 方言的转换,协调用户应用与远程数据源间的存取关系;
- 对用户(或开发者)屏蔽有关查询和使用远程数据源的细节;
- 使桌面应用 API 和后端协议公开。

学术界和产业界都为此做了大量的工作,目前已推出一些可互操作的数据库平台产品。从体系结构上可以把它们大体归纳成四类:数据库访问 API;带常规信关的 RDBMS;开放式信关;数据库封装软件。这四种体系结构都不同程度地满足了数据库互操作性的要求,各有侧重,也各有利弊。下面分别加以讨论。

2 数据库访问 API

SAG (SQL Access Group), Microsoft 公司及 Borland 公司分别制定了一套访问多个后端数据源的应用编程界面,称为数据库访问 API。该 API 定义了应用编程的统一界面标准,独立于 DBMSs。它是为开发者提供的一套命令集,用于与远程数据库建立连接,并向它们发送 SQL 语句。制定该 API 的目的是,使开发者可以不依赖于 DBMS 厂商的应用工具,按标准的方法开发自己的访问后端数据库的 PC (或者 Macintosh, 或者 UNIX 工作站) 应用程序,从而代替现有的一些数据库前端应用软件,如 Lotus 的 DataLens, 先锋软件公司的 Q+E, SYBASE 的 DB-lib 等。其体系结构如图1所示。

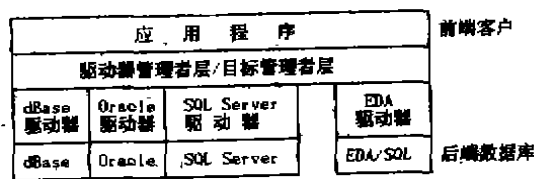


图1 数据库访问 API 的体系结构

数据库访问 API 为开发者提供了调用级的界面。应用程序中所调用的库函数实现对 SQL 语句的语法检查以及数据类型的检查,将检查后的结果移

*)本文得到电子科技大学青年科学基金和电科院基金的资助。

交驱动器管理者(或目标管理者);驱动器管理者确定相应目标数据库的驱动器,并执行初始化;然后由驱动器把目标数据库的逻辑名映射成数据库的实际名,完成客户 API 到服务器 API 的转换,发送 SQL 请求,提供事务服务以及与网络通信协议的接口。当服务器响应后,驱动器再把结果返回给用户。

数据库访问 API 实际上只解决了客户应用程序如何向后端数据库提交数据库服务请求的问题,对提出请求到返回结果这一过程所必需的网络服务却没有足够的描述。另外,API 性能的好坏完全取决于数据库厂商对访问目标数据库所需驱动器的实现。每个后端数据库都需要一个数据库驱动器。大多数的驱动器都是由 DBMS 厂商编写,厂商很可能在驱动器中只是有选择地支持 API 所定义功能的一部分,而有的 RDBMS 的驱动器还要求在客户应用与后端数据库间采用厂家自己的 SQL 连接软件来传输请求。这样,用户就不得不需要多种数据库连接层来支持多数据库互操作环境,既受到厂商的控制又增加了系统的成本。

目前较著名的数据库访问 API 有:SQL Access Group(简称 SAG)的嵌入式 SQL 界面和调用级界面,Microsoft 的开放式数据库连接 API(ODBC API),Borland 的开放式数据库应用编程界面(ODAPI)。

3 带常规信关的 RDBMS

带常规信关的 RDBMS 是较广泛采用的一种体系结构。它的提出主要是基于这样一种思想,即在现有某种关系数据库系统及其应用工具的支撑下,实现对其他后端数据库的访问。这就需要在该数据库系统和后端数据库系统之间提供一种中间纽带——信关,来达到动用目标数据库的目的。该关系数据库系统加上相应的信关,便构成了一个简单的可互操作的数据库平台。其结构如图2所示。

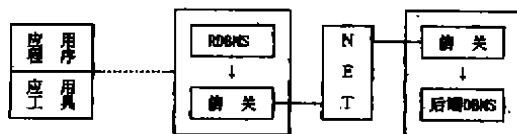


图2 带常规信关的 RDBMS 体系结构

RDBMS 不仅保存了自身数据库的所有信息,同时还保存着被连后端数据库的路径,对它们提供目录服务和名字服务,以及对它们的分布式查询及更新进行管理。用户通过系统所提供的应用工具发

出数据请求,由 RDBMS 对用户应用中的 SQL 语句进行分析,并根据所保存的外部数据库的信息,确定用户欲访问数据库的路径和位置,提供 SQL 路由选择,然后激活相应的数据库信关。由信关完成数据类型、数据格式、错误信息等转换,并提供与后端数据库匹配的 API。RDBMS 的信关在功能上类似前面我们讨论过的驱动器,只是二者的结构不同。典型的信关实际上是一个运行在与后端数据源相同机器上的特殊应用程序,并且在服务器 RDBMS 上有一个相应的拷贝。其功能一般是固定的,不能被扩展。Oracle, Ingres 及 Informix 都提供了基于带常规信关 RDBMS 的可互操作的数据库平台,具体如 Oracle * Connet, Ingres/Star 及 Informix * Star。

尽管带常规信关的 RDBMS 这种体系结构建立了一个比较透明且稳定的可互操作的数据库平台,但是在以下几方面仍表现出其局限性:

- 资源冗余。每一种外部数据库都需要一个信关,在服务器 RDBMS 上就得有一个相应的拷贝。
- 在网络协议及支撑信关方面受数据库厂商的制约,一般不提供数据库更新功能。
- 客户受限于某个 RDBMS 厂商,不能任意选择客户平台及后端数据库。

总之,带常规信关的 RDBMS 这种体系结构显得有些过时。尽管积极倡导此体系结构的数据库厂商不断完善,增加其可互操作的数据库平台的性能和功能,但因为传统信关的功能一旦由厂商确定,就无法动态地改变,除非厂商再进一步地对它们进行扩充。因此,功能有限、封闭式的信关、以及对某一种数据库系统的依赖,都将成为它达到真正数据库互操作性要求的障碍。

4 开放式信关

前面的三种体系结构尽管都能实现可互操作的数据库平台,但是在许多方面还存在着局限性,离数据库互操作性的要求还有很大的距离。一种在传统带常规信关的 RDBMS 体系结构上发展起来的新的实现可互操作数据库平台的体系结构——开放式信关,已受到人们的关注。

开放式信关为前端应用和后端数据库提供了中间纽带,但并不像带常规信关的 RDBMS 那样还需要某个 RDBMS 的支持。它是一种可以独立于服务器上数据库管理系统的体系结构。也就是说,开放式信关可以代替服务器上的 DBMS,透明地支持用户对后端数据库的访问。典型的开放式信关应提供:客户与后端数据库的透明连接功能;请求对目标数据

库的数据进行操作的路由选择功能,以及名字服务和 SQL 翻译等功能。一般包括开放式客户和开放式信关两部分,前者即所谓客户应用编程界面,后者即数据库驱动器。其结构如图3所示:

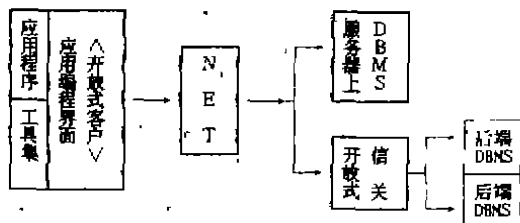


图3 开放式信关体系结构

与传统信关相比,开放式信关更具可扩展性。不仅允许多种客户方工具及应用程序使用它们,而且在服务方还采用了驱动器方式的结构来支持多种后端 DBMS。此外,大多数开放式信关还提供 API,用于构造客户应用以及为新的数据源构造新的驱动器。事实上,开放式信关使得开发者能够构造连接桌面应用和一系列专门的后端服务(如图像管理和股票市场分析等)的信关。

目前, Sybase 和 Progress Software 公司是推崇开放式信关的主要数据库厂商,其他还有 Information Builders 公司。而 IBM 的 DRDA(分布式关系数据库体系结构)也被认为是一个开放式信关,但它是专门为 IBM 的几个 RDBMS 开发的。下面以 Sybase 的 Open Client/Server 为例,来说明开放式信关的实现。Sybase 的 Open Client/Server 是第一个较成熟的开放式信关产品。它采用了调用级的 API 提供与外部关系型或非关系型数据库的连接,如 DB2, Oracle, Informix 等。在客户方, Sybase 提供了公开的 DB-lib 和 NET-lib 库,允许开发者和第三方厂商以函数调用的方式开发访问后端数据库的前端应用工具。基于 DB 库的典型应用是 Power Builder。在服务方, Server-lib 及 NET-lib 又为开发者提供了构造新的数据库驱动器的手段,以实现非 SYBASE 数据源的共享。另外, Sybase 还支持对分布式数据库进行更新的两阶段提交协议,以及支持包括 IBM DRDA, DECnet, Borland ODAPI & OSF DCE 等在内的多种网络协议^[11-12]。

开放式信关可看成是一种吸收了前端 API 和常规信关的优点,采用驱动器的方式灵活地支持多种后端数据源的体系结构。它不仅能较好地实现对后端数据库的透明访问,支持多种网络协议和多种客户机平台,更重要的是其开放的应用编程界面,为

开发者和第三方厂商提供了扩充客户应用工具和信关的手段。其缺点是增加了建立开放式信关所需数据定义的开销及冗余,因为要提供对新的数据源的支持,开发者必须借助 API 编写大量的构造新数据库驱动器的程序,而对 API 库函数的依赖又容易受到厂商的制约。

5 数据库封装

数据库封装采用了面向对象的技术,把后端数据库用对象的机制封装起来,实现应用程序对远程多数据库的访问。在数据库封装中,对象描述了数据库对应用程序提供的存取功能以及具体数据信息。用户通过调用数据库对象来使用不同的数据库资源,这样,所有数据库看起来就好像是处于一个统一的应用环境中。因此,封装为异种数据库提供了一个统一的数据模式,掩盖了不同数据库间的差异。数据库封装不仅支持数据库的读写访问,并行操作,而且还支持数据库特有功能的访问,几乎能满足数据库互操作性的所有特征。毫不夸张地说,数据库封装是目前最有效的一种可互操作数据库平台的体系结构。

下面我们以一种基于 RPC(远程过程调用)的数据库封装为例^[3],简要地说明一下数据库封装中对对象的处理。在服务方要求提供定义数据库对象的功能,通常采用扩充标准 SQL 语言的方法,即增加有关对象定义的语句。对象定义包括属性、操作及约束条件的定义。而属性定义又具体定义了数据库的域名、类型、大小。操作则定义了与数据库的接口。在客户方,则为用户提供两种界面,一种是对 SQL 的扩充,增加访问对象的命令;另一种是嵌入式的 API,即提供可以嵌入 C 或 C++ 高级语言的库函数。这些库函数利用 RPC 机制完成对数据库对象的访问,从而达到访问目标数据库的目的。

数据库封装使用户对数据库的访问不再是直接的,而是通过对象进行间接存取,这在很大程度上减少了数据库变化对现存应用程序造成的影响。其次,开发者可以通过定义对象的不同操作,实现数据库的各种功能,提交给用户使用。数据库封装能够很容易地支持多通信协议和多协议转换;通信协议可以作为数据库对象的一种属性进行定义,而协议转换则定义成对象的一个操作。利用对象的灵活性,数据库封装还为开发者提供了一个开发数据访问应用程序的环境;开发者可以开发满足自己需要的数据访问应用。另外,利用对象的继承性,还能够很容易地

(下转第57页)

3.2 体系结构

LNFDDBS 的体系结构如图5:

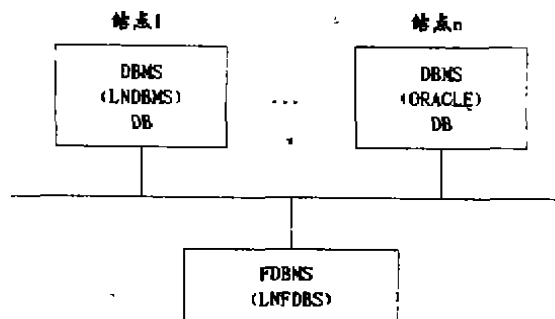


图5

LNFDDBS 采用类似于总线形网络的体系结构,各个结点相当于工作站,FDBMS 相当于服务器,各个结点的 DBMS 具有自治性,同时也可通过 FDBMS 访问其它结点。当然,更重要的是,它也允许任意结点之间的直接通信。这是必要的,因为 FDBMS 结点有可能成为瓶颈,影响系统效率,此外这种结构也更加适用于企业的组织模式。

目前,我们以 ORACLE、FOXBASE (FOX-PRO)、LNDBMS 作为 LNFDDBS 中的三个结点,ORACLE 是一个功能很强的通用数据库系统,它的安全性和完整性都很强,但它的效率在一般的微机资源下不很高^[10],FOXBASE (FOXPRO) 的效率较高,但安全性却很差^[10],LNDBMS 是我们正在开发的面向数据采集的专用数据库系统,它具有很高的效率和较好的安全性^[10,11]。我们选择它们就是想综合它们的优点,使 LNFDDBS 既有很高的效率和较好的安全性,又有很强的功能。

本文介绍了 LNFDDBS 的特点和体系结构,有关

其进一步的技术问题,将另文发表。

参考文献

- [1] 唐雪飞,数据库的互操作性,计算机科学,Vol. 20, No. 3, 1993
- [2] W. Litwin, et al., Interoperability of Multiple Autonomous Database, ACM computing Surveys, Vol. 22, No. 3, 1990
- [3] W. Litwin, et al., Multi-database interoperability, IEEE Comput. Vol. 19, No. 12, 1986
- [4] S. Ceri, et al., Distributed Databases-Principles and System, McGraw-Hill, New York, 1984
- [5] W. Litwin, et al., SIRIUS System for Distributed Data Management, in Distributed Data Bases, H.-J. Schneiders, Ed. 1982
- [6] K. Hammer, et al., On database management system architecture, Tech. Rep. MIT/LCS/TM-144, 1979
- [7] D. Heimbigner, et al., A Federated Architecture for Information Management, ACM Trans. on Inf. Syst. Vol. 3, No. 3, 1985
- [8] A. P. Sheth, et al., Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, ACM Computing Surveys, Vol. 22, No. 3, 1990
- [9] S. Ram, et al., Architecture of Distributed DataBase System, the Journal of System and Software, Vol. 10, No. 2, 1989
- [10] 肖永桥,姚卿达,单一式和客户/服务器数据库系统:测试和分析,计算机科学,Vol. 21, No. 5, 1994
- [11] 姚卿达,肖永桥,面向数据采集的专用数据库管理系统—LNDBMS 的设计与实现(待发表)

(上接第54页)

建立高层数据视图。

数据库封装的缺点主要表现在对厂商所提供封装软件的依赖上,因为,对数据库进行封装要求专门的技术和编程技能,通常厂商对数据库封装框架是全封闭的,不支持第三方厂家提供新的数据库封装手段;同时,厂商还必须提供自己的 API 和前端应用,允许客户机上的应用可以对数据库对象发出请求,往往这种 API 较难掌握,与 SAG 或者 Microsoft 的 API 相比,显得有此曲高和寡。尽管还存在一定的缺点,但随着面向对象技术的逐渐成熟,数据库封装将成为现阶段解决数据库互操作性,实现可互操作数据库平台最有希望、最有前景的体系结构。目前主要的数据库封装软件厂家有 Constellation Software 公司, OpenBooks Software 公司及 Ontos 公司。

6 结论

通过对上述五种体系结构的讨论,我们可以得出这样的结论:开放式信关和数据库封装是目前较理想的互操作数据库平台的体系结构,开放式信关以 SQL 语言为基础,对用户来说,实现现有的各种数据库互操作性是可以获得的,事实上 SYBASE 在这方面已取得了很大的成功,尽管基于数据库封装方法的互操作数据库平台产品还不成熟,但是所采用的面向对象的技术本身就意味着数据库封装将是更灵活、更长期的解决数据库互操作的途径。由于能够较好地解决一种数据库系统与多数据库系统的透明访问,带常规信关的 RDBMS 体系结构仍将继续为人们所接受。至于数据库访问 API,笔者认为,很有希望与开放式信关及数据库封装的客户 API 相结合,成为一种互操作数据库平台的应用界面的标准。(参考文献共14篇略)